

日本国特許庁
JAPAN PATENT OFFICE

10/028823
12/26/01
PRO
979 U.S.



別紙添付の書類に記載されている事項は下記の出願書類に記載され
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日

Date of Application:

2001年 5月 25日

出願番号

Application Number:

特願 2001-156289

出願人

Applicant(s):

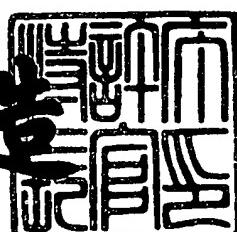
株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年10月26日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願
【整理番号】 K01007791A
【あて先】 特許庁長官殿
【国際特許分類】 G06F 12/00
【発明者】
【住所又は居所】 神奈川県川崎市幸区鹿島田 890 番地 株式会社日立製作所 ビジネスソリューション事業部内
【氏名】 星野 慎一
【発明者】
【住所又は居所】 神奈川県川崎市幸区鹿島田 890 番地 株式会社日立製作所 ビジネスソリューション事業部内
【氏名】 原 憲宏
【特許出願人】
【識別番号】 000005108
【氏名又は名称】 株式会社日立製作所
【代理人】
【識別番号】 100075096
【弁理士】
【氏名又は名称】 作田 康夫
【手数料の表示】
【予納台帳番号】 013088
【納付金額】 21,000円
【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 データベース管理方法およびシステム並びにその処理プログラムおよびその処理プログラムを格納した記録媒体

【特許請求の範囲】

【請求項1】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成し、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムを起動する際に、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定し、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とし

上記プログラムの終了時に、上記プログラムの終了時まで使用不可としたリソースを使用可とすることを特徴とするデータベース管理方法。

【請求項2】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成し、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムの起動要求を出力し、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定し、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とし

上記プログラムの終了時に、上記プログラムの終了時まで使用不可としたリソースを使用可とすることを特徴とするデータベース管理方法。

【請求項3】

上記トランザクション解決要求は、コミットステートメントを含むことを特徴とする請求項1から請求項2のいずれかに記載のデータベース管理方法。

【請求項4】

上記リソースへのアクセス不可時に当該リソースの識別情報を保持し、

上記他のプログラムからのアクセス要求を入力した場合、該アクセス要求が上記保持されたリソースの識別情報であった場合は、アクセスを不可とすることを特徴とする請求項1から請求項2のいずれかに記載のデータベース管理方法。

【請求項5】

上記アクセスを不可としたリソースを可とするとき、上記保持したデータベースリソースの識別情報を削除することを特徴とする請求項4に記載のデータベース管理方法。

【請求項6】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成する手段と、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムを起動する際に、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定する手段と、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とする手段と、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とする手段とを備えたことを特徴とするデータベース管理システム。

【請求項7】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成する手段と、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムの起動要求を出力し、上記プログラムでの使用が予め決定されたリソースについて、

他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定する手段と、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可する手段と、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とする手段とを備えたことを特徴とするデータベース管理システム。

【請求項8】

上記トランザクション解決要求は、コミットステートメントを含むことを特徴とする請求項6から請求項7のいずれかに記載のデータベース管理システム。

【請求項9】

上記リソースへのアクセス不可時に当該リソースの識別情報を保持する手段と

上記他のプログラムからのアクセス要求を入力した場合、該アクセス要求が上記保持されたリソースの識別情報であった場合は、アクセスを不可とする手段とを備えたことを特徴とする請求項6から請求項7のいずれかに記載のデータベース管理システム。

【請求項10】

上記アクセスを不可としたリソースを可とするとき、上記保持したデータベースリソースの識別情報を削除する手段を備えたことを特徴とする請求項9に記載のデータベース管理システム。

【請求項11】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成するステップと、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムを起動する際に、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定するステップと、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とするステップと、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とするステップとを有することを特徴とするデータベース管理プログラム。

【請求項12】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成するステップと、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムの起動要求を出力し、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定するステップと、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とするステップと、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とするステップとを有することを特徴とするデータベース管理プログラム。

【請求項13】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成するステップと、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムを起動する際に、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定するステップと、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とするステップと、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とするステップとを有することを特徴とするデータベース管理プログラムを記録した計算機読み取り可能な記憶媒体。

【請求項14】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成するステップと、

上記実行モジュールの実行時、該実行モジュールに含まれないプログラムの起動要求を出力し、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定するステップと、

上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とするステップと、

上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とするステップとを有することを特徴とするデータベース管理プログラムを記録した計算機読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データベース管理技術に関し、処理要求に応じてプログラムを起動してデータベース処理要求を行う際のデータベースリソースを管理する技術に関するものである。

【0002】

【従来の技術】

データベース言語SQLは、ユーザが定義した関数およびプロジェクトをルーチンとして呼出す機能を持つ。上記関数およびプロジェクトは、ルーチンの定義時に解析され、ルーチン名と関連付けられた実行モジュールとして登録される。実行モジュールとは、一連のデータベース処理手順を解析し、データベース管理システム内で実行できる形態にしたものである。ルーチン呼出し時にはルーチン名を指

定することで、登録された実行モジュールに従いデータベース処理が行われる。

【0003】

一方、文献(Andrew Eisenberg and Jim Melton, SQLJ - Part 1 : SQL Routines using the Java Programming Language, SIGMOD Record, Vol.28, No.4, December 1999)では、ルーチンとしてJava言語により記述されたプログラムを実行する機能について述べている。(Javaは、米国Sun Microsystems, Incの登録商標である)

この機能では、ルーチン定義時にプログラムがルーチン名と関連付けて登録される。ルーチン呼出し時には、データベース管理システムによりプログラム実行環境上でプログラムが実行される。プログラムからSQLステートメントが発行される際には、プログラム実行環境がデータベース管理システムを呼び返すことで実行モジュールが作成され、データベース処理が行われる。

【0004】

【発明が解決しようとする課題】

発行するSQLステートメントがルーチン呼出しによるプログラム実行時に決定する環境下では、実行モジュールが実行時に作成され、SQLステートメント実行のためのデータベースリソースも実行時に決定される。データベースリソースとは、SQLステートメント実行のために必要なメモリ領域およびSQLステートメント実行のために確保するカーソル、ロックなどである。データベースリソースの中には、トランザクション解決が行われるまで確保すべきものがある。トランザクションとは、データベース回復のための最小単位として保証される一連のSQL操作である。commitまたはrollback要求にてトランザクション解決が行われ、トランザクション中に発行したSQLステートメントによるデータベース更新結果が保証または取り消される。

【0005】

上記環境下では、プログラム実行中に発行されるSQLステートメントに関するデータベースリソースは、ルーチン呼出し用の実行モジュールの制御下にない。このため、データベースリソースは、ルーチン呼出し側で行っていたトランザクション解決に応じて管理していた。

【0006】

ルーチン呼出しによって実行されるプログラム中でトランザクション解決を可能にすると、クライアント側で実行していたトランザクションを含むプログラムを、そのままルーチンとして実行することが可能となる。このためには、上記で示した、プログラム実行中に発行されるSQLステートメントに関するデータベースリソースの解放手段について考慮することが課題となる。

【0007】

発明(U.S.P. 6006235)は、Java言語などのインタプリタ言語により記述されたプログラムをルーチンとして実行するための手法、手段に関して述べているが、プログラム実行中のトランザクション解決に関しては述べられていない。

【0008】

本発明の目的は、上記課題を改善するために、他のプログラムで使用できるデータベースリソースを好適に解放するデータベース管理方法及びシステムを提供することにある。

【0009】

【課題を解決するための手段】

上記課題を改善するために本発明では、請求項1もしくは請求項2の処理を行う。

これにより、他のプログラムで使用できるデータベースリソースを好適に解放することが可能となる。

【0010】

【発明の実施の形態】

プログラム中からのトランザクション解決が可能な一実施形態のデータベース管理システムについて説明する。

まず、本発明の概念について図1を用いて簡単に説明する。

【0011】

本実施形態のデータベース管理システムにおけるルーチン実行は、図1に示すように、SQL実行環境100、プログラム実行環境101、リソース確保部102、リソース解放部103、プログラム実行制御部104により行われる。ユーザからのルーチン

呼出し要求に伴い、SQL実行環境100は、実行モジュール130を作成し実行する。ディクショナリ情報格納箇所110内のディクショナリ120に対し、ルーチンとプログラムとの関連情報を参照し、起動するプログラムを決定する。SQL実行環境100は、リソース確保部102に対しリソース種別150をパラメタとしてルーチン呼出しに伴うリソース140の確保要求を行う。リソース確保部102は、リソース種別150を解放リソース管理テーブル格納箇所111内の解放リソース管理テーブル121およびリソース管理テーブル格納箇所112内のリソース管理テーブル122に登録する。上記処理により、リソース種別150は、現在確保されているリソースであり、かつ、ルーチン呼出し終了まで解放不可であるリソースとして登録される。本発明における確保とは、リソースを割り当てたルーチン（プログラムともいう）にリソースのアクセス権限を与え、他のルーチンやプログラム実行制御部には同じアクセス権限を与えないというものである。例えば、書き込み権限の場合には、リソースが確保されていないプログラムからの書き込みはできることになる。

【0012】

ディクショナリより取得した関連情報に基づき、SQL実行環境は、プログラム実行制御部104に対し起動要求を行う。プログラム実行制御部104は、プログラム実行環境101に対しプログラム起動を行う。プログラム133がSelect文134を発行する場合、プログラム実行環境101は、SQL実行環境100にSelect文134の実行要求を行う。SQL実行環境100は、実行要求に基づき実行モジュール131を作成し実行する。SQL実行環境100は、リソース確保部102に対しリソース種別151をパラメタとしてSelect文実行に伴うリソース141の確保要求を行う。リソース確保部102は、リソース種別151をリソース管理テーブル格納箇所112内のリソース管理テーブル122に登録する。上記処理により、リソース種別151は、現在確保されているリソースとして登録される。

【0013】

プログラム133がCommit135を発行する場合、プログラム実行環境101は、SQL実行環境100にCommit135の実行要求を行う。SQL実行環境100は、実行要求に基づき実行モジュール132を作成し実行する。SQL実行環境100は、リソース解放部103に対し、リソースの解放要求を行う。リソース解放部103は、解放リソース管理テ

ーブル格納箇所111内の解放リソース管理テーブル121およびリソース管理テーブル格納箇所112内のリソース管理テーブル122を参照し、解放可能なリソース種別152を決定する。この決定は、リソース管理テーブル112に登録されているリソースで解放リソース解放テーブル111に登録されていないリソースを抽出することで行う。本例では、T1とI1がリソースを解放して良いことが分かる。リソース解放部103は、リソース管理テーブル格納箇所112内のリソース管理テーブル122から、リソース種別152を削除する。上記処理により、リソース種別152は、解放される。

【0014】

次に、図2に本実施形態のデータベース管理システムの機能ブロックおよびハードウェア構成を示す。

コンピュータシステム200は、CPU210、主記憶装置211、磁気ディスク装置などの外部記憶装置202および多数の端末201で構成される。主記憶装置211上には、データベース管理システム220およびプログラム実行環境101が置かれる。外部記憶装置202上には、データベース管理システム220が管理するデータベース251およびディクショナリ252が格納される。また、データベース管理システム220およびプログラム実行環境101を実現する処理プログラム250も、外部記憶装置202上に格納される。

【0015】

データベース管理システム220は、システム制御部230、SQL実行環境100、プログラム実行制御部104、データベースアクセス処理部231、ディクショナリアクセス処理部232で構成される。システム制御部230は、リソース確保部102、リソース解放部103、解放リソース管理テーブル121、リソース管理テーブル122で構成される。

端末201よりユーザが発行するSQL実行要求は、SQL実行環境100にて実行モジュール240、241として実行される。要求されたSQLステートメントがルーチン呼出しの場合、SQL実行環境100は、ディクショナリアクセス処理部232に対しルーチンと実行するプログラムの関連情報取得要求を行う。ディクショナリアクセス処理部232は、外部記憶装置202内のディクショナリ252に対し関連情報の参照を行

い、SQL実行環境100に返却する。SQL実行環境100は、上記関連情報に基づき、プログラム実行制御部104に対しプログラム実行要求を行う。プログラム実行制御部104は、プログラム実行環境101にてプログラム242を実行する。

【0016】

システム制御部230内のリソース確保部102は、SQL実行環境100でSQLステートメント実行時に必要となるリソースを確保するため、解放リソース管理テーブル121およびリソース管理テーブル122にリソース種別を登録する。システム制御部230内のリソース解放部103は、SQL実行環境100からリソース解放要求を受けた際に、解放リソース管理テーブル121およびリソース管理テーブル122を参照して解放するリソースを決定し、リソース管理テーブル122から決定したリソース種別を削除することでリソースの解放を行う。

図3は図1における本実施形態のSQL実行環境が、ユーザから要求されたルーチン呼出し用の実行モジュール130を実行し、プログラム実行環境101上でプログラム133を実行する処理手順を示すフローチャートである。ステップ300～305はSQL実行環境100、ステップ310はリソース確保部102、ステップ320はリソース解放部103、ステップ330はプログラム実行環境101で行われる処理である。

【0017】

まず、ステップ301において、SQL実行環境100は、ディクショナリ120を参照して、ルーチンとプログラムの関連情報を取得する。次に、ステップ302において、SQL実行環境100は、リソース確保部102にルーチン呼出しに伴うリソースの種別150をパラメタとして渡し、リソースの確保要求を行う。これに伴いステップ310において、リソース確保部102は、パラメタとして取得したリソース種別150を解放リソース管理テーブル121およびリソース管理テーブル122に登録する。この結果、解放リソース管理テーブル121およびリソース管理テーブル122は、340に示すように、リソース種別150として「定義R1」「プログラムP1」を保持した状態となる。

そして、ステップ303において、SQL実行環境100は、ステップ301で取得した関連情報から起動するプログラムを決定し、プログラム実行制御部にプログラム実行要求を行う。これに伴いステップ330において、プログラム実行環境101は指定さ

れたプログラムを実行する。ステップ330で示すプログラム実行処理の詳細は、図4にて説明する。

【0018】

プログラム実行処理が終了した後、ステップ304において、SQL実行環境100は、リソース解放部103にルーチン呼出しに伴うリソースの種別150をパラメタとして渡し、リソースの解放要求を行う。これに伴いステップ320において、リソース解放部103は、パラメタとして取得したリソース種別150を解放リソース管理テーブル121およびリソース管理テーブル122から削除する。この結果、解放リソース管理テーブル121およびリソース管理テーブル122は、341に示すように、リソース種別150が削除され何も保持していない状態となる。上記手順でSQL実行環境100でのプログラム呼出し処理が行われる。

【0019】

図4は図3のステップ330の詳細として、図1における本実施形態のプログラム実行環境101が、プログラム133の実行処理を行い、Select文134およびCommit135を発行し、SQL実行環境100を呼び返す処理手順を示すフローチャートである。ステップ410～413およびステップ430～432はSQL実行環境100、ステップ420はリソース確保部102、ステップ440はリソース解放部103、ステップ400～403はプログラム実行環境で行われる処理である。

【0020】

まず、ステップ401において、プログラム実行環境101は、プログラムからSQLステートメントが発行されているかどうかを判定する。SQLステートメントが発行されている場合、ステップ410に進みSQL実行処理を開始する。ステップ411において、SQL実行環境100は、発行されたSQLステートメントを解析する。解析により、SQLステートメント実行に必要なリソースの種別151を決定する。ステップ412において、SQL実行環境100は、リソース確保部102に、上記のリソース種別151をパラメタとして渡し、リソースの確保要求を行う。これに伴い、ステップ420において、リソース確保部102は、パラメタとして取得したリソース種別151をリソース管理テーブル122に登録する。この結果、リソース管理テーブル122は、450に示すように、図3における状態340にリソース種別151として「表T1」「イン

「インデックスI1」が追加され、「定義R1」「プログラムP1」「表T1」「インデックスI1」の4つを保持した状態となる。以上ステップ411、412によりSQL実行環境100は、SQLステートメント実行処理を終了しプログラム実行環境101に返る。

【0021】

ステップ401において、SQLステートメントが発行されていない場合、ステップ402に進む。ステップ402において、プログラム実行環境101は、プログラムからCommitが発行されているかどうかを判定する。Commitが発行されている場合、ステップ430に進みCommit処理を開始する。ステップ431において、SQL実行環境100は、リソース解放部103に、リソースの解放要求を行う。これに伴い、ステップ440において、リソース解放部103は、解放リソース管理テーブル121およびリソース管理テーブル122を参照して解放するリソース種別152を決定する。ここでは、状態450においてリソース管理テーブル122に保持されており、かつ、解放リソース管理テーブル121に保持されていないリソース「表T1」「インデックスI1」が、解放するリソース種別152として決定される。そして、リソース解放部103は、上記のリソース種別152をリソース管理テーブル122から削除する。この結果、リソース管理テーブル122は、451に示すように、状態450からリソース種別151として「表T1」「インデックスI1」を削除し、「定義R1」「プログラムP1」のみを保持した状態となる。以上ステップ431によりSQL実行環境100は、Commit処理を終了しプログラム実行環境101に返る。

【0022】

ステップ402において、Commitが発行されていない場合、プログラム実行処理を終了する。上記手順でプログラム実行環境101でのプログラム実行処理は行われる。

以上示したフローチャートの処理は、図2で示したコンピュータシステム200におけるプログラムとして実行される。しかし、そのプログラムは図2で示したようにコンピュータシステム200に物理的に接続された外部記憶装置に格納されるものとは限定しない。コンピュータシステム200に物理的に接続されていない、ハードディスク装置、フロッピーディスク装置などのコンピュータシステム200で読み書きできる記憶媒体に格納してもよい。また、ネットワークを介して図2

のコンピュータシステム200とは別のコンピュータシステムに接続された外部記憶装置に格納してもよい。

【0023】

上述の実施の形態においては、解放不可であるデータベースリソースを解放リソース管理テーブル121として保持している。ここで、解放リソース管理テーブル121を使用しないで、リソース管理テーブル122に解放不可状態を示す情報を追加することで、解放不可であるデータベースリソースの情報としてもよい。

本発明は、コミットで述べたがロールバックでも同様に適用できる。

【0024】

本発明によれば、プログラム実行中にトランザクション解決を行えるようにすることで、データベース処理を行うユーザにとっては、トランザクションを含むプログラムをそのままルーチン呼び出し用プログラムとして使用できるため、可用性が向上する。

【0025】

【発明の効果】

本発明によれば、他で使用できるデータベースリソースを好適に解放することが可能となる。

【図面の簡単な説明】

【図1】本発明の概念図である。

【図2】本実施形態のデータベース管理システムの機能ブロックおよびハードウェア構成を示す図である。

【図3】本実施形態のSQL実行環境が、プログラム実行環境上でプログラムを実行する処理手順を示すフローチャートである。

【図4】図3のステップ330の詳細として、本実施形態のプログラム実行環境が、プログラムの実行処理を行い、SQL実行環境を呼び返す処理手順を示すフローチャートである。

【符号の説明】

100…SQL実行環境、101…プログラム実行環境、102…リソース確保部、103…リソース解放部、104…プログラム実行制御部、120…ディクショナリ、121…リソ

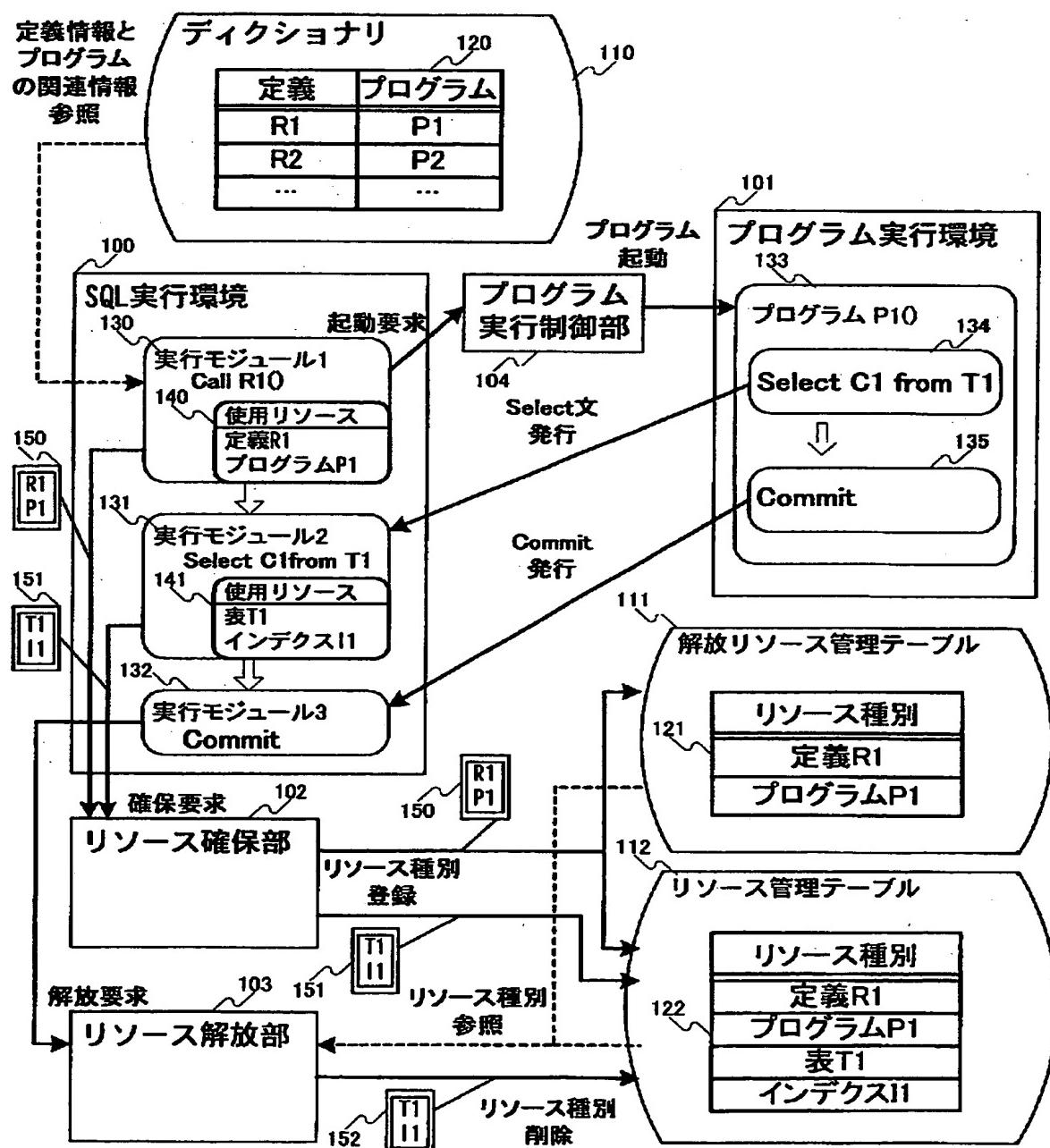
特2001-156289

ース管理テーブル、122…解放リソース管理テーブル、220…データベース管理シ
ステム、230…システム制御部、231…データベースアクセス処理部、232…ディ
クショナリアクセス処理部、251…データベース

【書類名】 図面

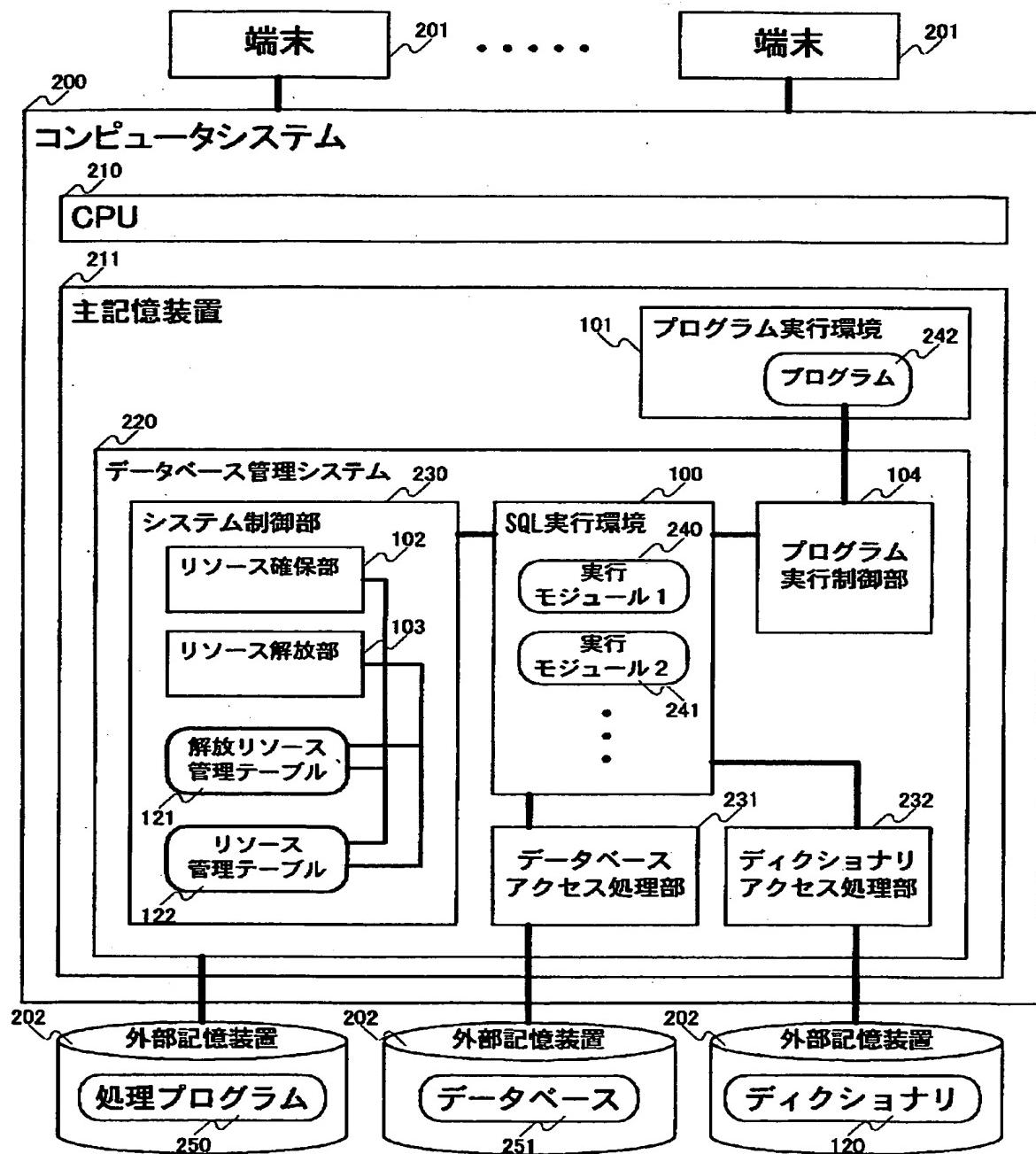
【図1】

図1



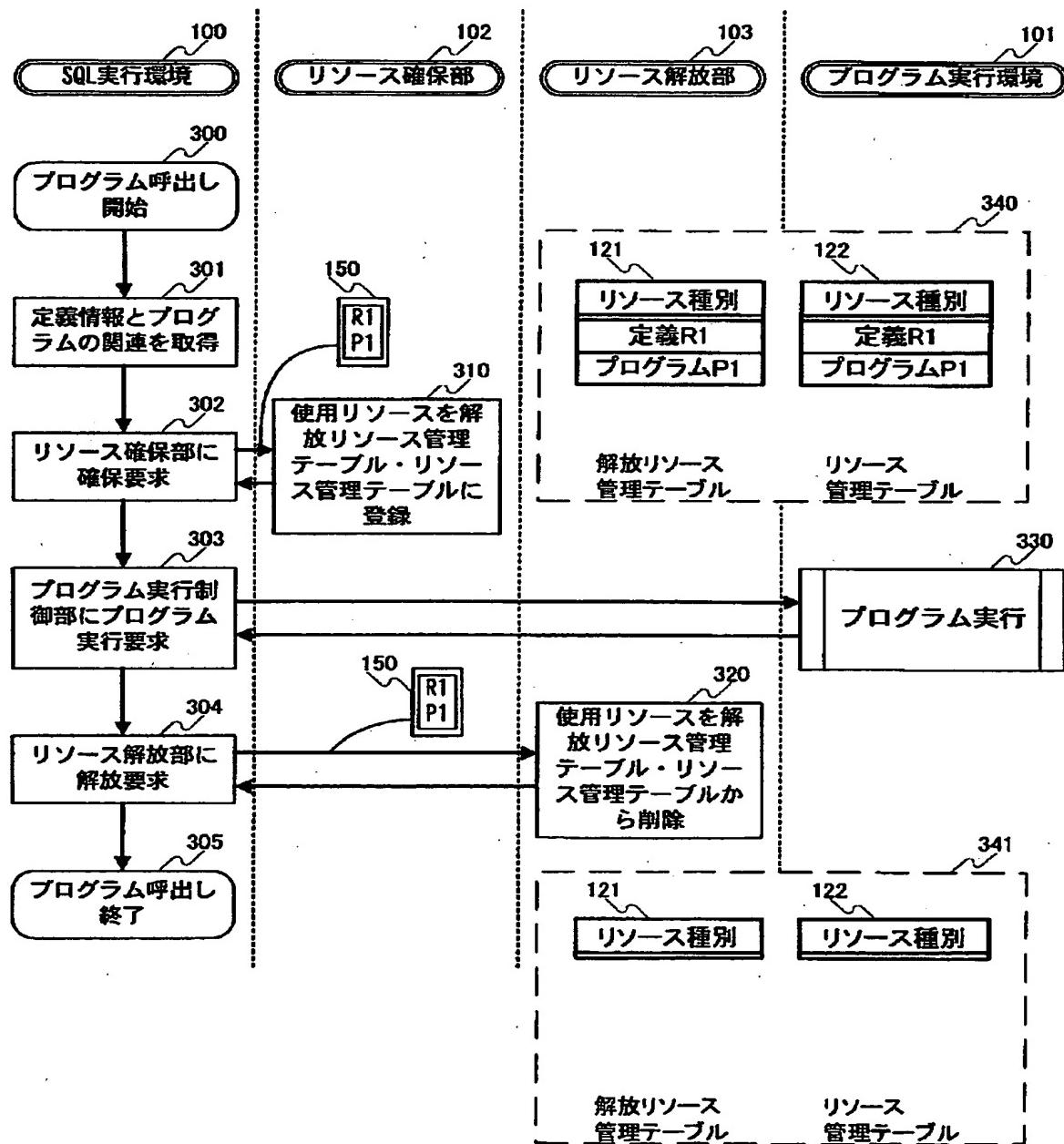
【図2】

図2



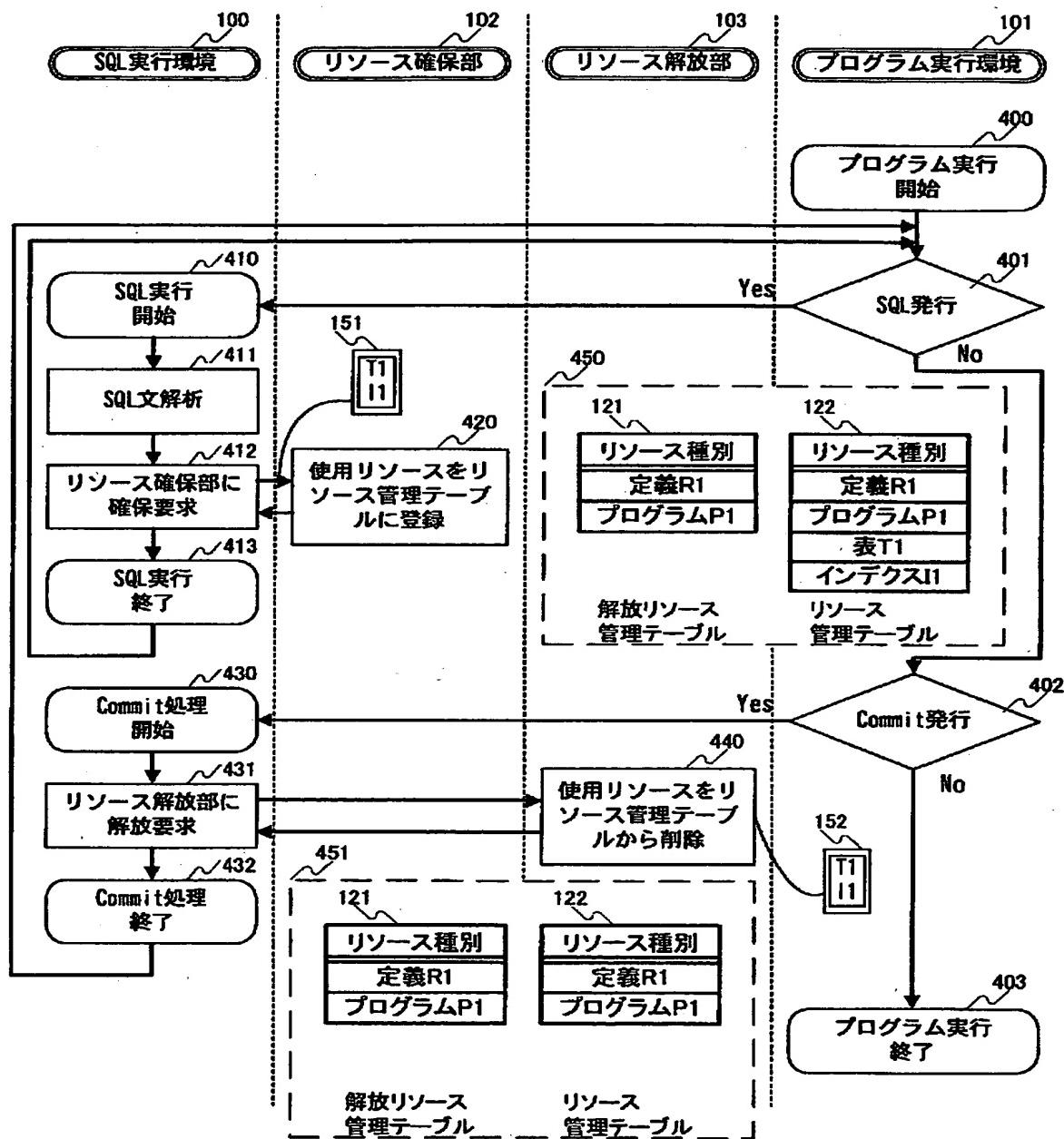
【図3】

図3



【図4】

図4



【書類名】 要約書

【要約】

【課題】

データベースに予め定義登録しておいたプログラムを処理要求に応じて起動してデータベース処理を行う際の、データベースリソースを効率的に管理する技術を提供するものである。

【解決手段】

入力した問い合わせ要求を解析して上記問い合わせ要求を実行するための実行モジュールを生成する。上記実行モジュールの実行時、該実行モジュールに含まれないプログラムを起動する際に、上記プログラムでの使用が予め決定されたリソースについて、他のプログラムにトランザクション解決要求時まで使用不可とするリソースと該他のプログラムの終了時まで使用不可とするリソースとを決定する。上記プログラムからトランザクション解決要求を入力したとき、上記他のプログラムにトランザクション解決要求時まで使用不可としたリソースを使用可とする。上記プログラムの終了時に、上記他のプログラムの終了時まで使用不可としたリソースを使用可とする。

【選択図】 図1

認定・付加情報

特許出願の番号	特願2001-156289
受付番号	50100751296
書類名	特許願
担当官	第七担当上席 0096
作成日	平成13年 5月28日

<認定情報・付加情報>

【提出日】 平成13年 5月25日

次頁無

出願人履歴情報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地

氏 名 株式会社日立製作所